Ideas

* Include (use Timestamp java function) to see when file created/modified into hash value
  + File file = new File("directory1/file1.txt");
  + System.out.println("Before Format : " + file.lastModified());
  + (requires import java.io.File;)
* Before making modifications check hash value to make sure they have the same file. If the values are different someone modified it
* FileInputStream that reads input (scanner) that corresponds with file in directory 1, hashes it, and outputs in directory 2.
  + Scanner scanner = new scanner(System.in);
  + String directory1 = scanner.next();
  + String file = scanner.nextLine();
  + try {
  + FileInputStream fis = new FileInputStream(directory1 + "file");
  + FileOutputStream fos = new FileOutputStream("directory2/encrypted.txt");
  + encrypt(key, fis, fos);
  + }
* Boolean value that tells us if hash value already exists or not.
* If value doesn’t exist, add file to directory
* If value matches when we don’t want it to there exist two problems:
  + Our algorithm is non-resistant collison and needs to be change
  + File wasn’t modified
* If value matches when we expect it to means value was not modified
* If values are different that means someone has modified it.

<http://www.mkyong.com/java/how-to-get-the-file-last-modified-date-in-java/>

KeyPairGenerator keyGen = KeyPairGenerator.getInstance("DSA");

SecureRandom random = SecureRandom.getInstance("SHA1PRNG", "SUN");  
keyGen.initialize(1024, random);

Questions to ask Prof. Samanthula tomorrow:

1. For project deliverable part (a), are there any requirements or suggestions for the format of the output file that gets saved in directory 2, for example, the hash of each file in directory 1 gets stored on a separate line of the output file?
2. Do we have to think about authentication?
3. Are there multiple files in directory 1?
4. Do files stay in directory 1 after they are hashed?
5. Should the program1 be run like:
   1. "java program1 *directory1* *directory2*", where both directories are command line arguments?

\*Test with multiple different file sizes

The integrity has been / has not been protected